

Projet du pôle Logique Computationnelle -
Prolog - Complexité - Décidabilité
Thème du projet:
Génération automatique d'emploi du temps

1 Contexte et problématique

L'objectif est d'utiliser la logique des prédicats, les notions de complexité et Prolog pour générer un emploi du temps.

L'ensemble des données intervenant pour la constitution d'un emploi du temps apparaît en appendice A. On y retrouve les différents éléments nécessaires à la génération d'un emploi du temps :

- promotions.
- professeurs.
- matières.
- groupes.
- séances.

Afin de gérer les contraintes permettant une génération des créneaux d'une semaine, nous disposons également des indisponibilités des professeurs ainsi que l'ensemble des informations associées aux groupes d'élèves et aux séances.

On dispose également d'informations permettant de rendre indisponibles certains créneaux (journées déscolarisées, jeudi après-midi), pour un groupe donné.

2 Base de données

Il est demandé de stocker l'ensemble de ces données dans autant de tables que nécessaires et de bien identifier les relations de dépendances.

1. Traduire l'ensemble des structures de données de l'Appendice A via le calcul des prédicats.
Exemple : la relation `matiere` peut être représenté par le simple prédicat `matiere(logique)`.
2. Traduire le jeu de test de l'appendice B en Prolog en suivant votre traduction des structures de données en calcul des prédicats.

LIVRABLE : La traduction en Prolog des données du jeu de test.

3 Interrogation de(s) base(s) de données

On doit pouvoir poser les questions suivantes :

1. Disponibilités d'un professeur pour une semaine donnée sous forme d'une liste de listes d'intervalles.

Exemple : ? `dispoProf(toto,12)`.

Une réponse est `[[8,18],[9,17],[8,18],[8,18],[9,16]]`, signifiant que la semaine 12, `toto` est disponible :

- (a) le lundi de 8h à 18h
- (b) le mardi de 9h à 17h
- (c) le mercredi de 8h à 18h
- (d) le jeudi de 8h à 18h
- (e) le vendredi de 9h à 16h

2. Disponibilités des créneaux d'une semaine donnée, sous forme d'une liste de listes d'intervalles.

Exemple : `dispoCreneaux(12)`.

Une réponse est `[[9,18],[8,18],[8,18],[8,12,5],[8,17]]`, signifiant que la semaine 12, les créneaux disponibles sont :

- (a) le lundi de 9h à 18h
- (b) le mardi de 8h à 18h
- (c) le mercredi de 8h à 18h
- (d) le jeudi de 8h à 12,5h
- (e) le vendredi de 8h à 17h

3. Liste des séances d'un groupe selon la semaine :

Exemple : `listeSeances(groupe1,12)`.

Une réponse est `[seance1,seance3,seance5,...]`, signifiant que la semaine 12 le groupe `groupe1` doit suivre les séances :

`seance1,seance3,seance5,...`

LIVRABLE : Les requêtes en Prolog basées sur votre modélisation des données.

4 Contraintes sur une liste de créneaux

L'objectif de cette section est de mettre en place un ensemble de contraintes permettant de spécifier des intervalles de valeurs à une liste de créneaux. Les créneaux d'un groupe pour une semaine seront modélisés par une liste croissante d'entiers de 1 (créneau correspondant au premier cours du Lundi) à n (dernier créneau de la semaine). On pourra par exemple découper les valeurs des créneaux en intervalles de 15 minutes : si le créneau $C = v$ équivaut à x h, le créneau $C' = v + 1$ équivaut à $x + 0,25$ h (Attention aux modulus pour les changements d'heure et de jour).

1. Exemple : Soit `ListeCreneaux=[C1,C2,...,CN]` une liste de créneaux. Le prédicat `all_different(ListeCreneaux)` contraint l'ensemble des créneaux à être différents. L'ensemble des outils disponibles pour fixer des contraintes est disponible via la bibliothèque `clpfd` que nous étudierons au chapitre 4 du cours de Prolog.
2. Écrire une contrainte qui intègre les disponibilités des professeurs pour une liste de séances.
3. Écrire une contrainte qui impose, pour une même matière, que les cours magistraux aient lieu avant les td/tp.
4. Écrire une contrainte qui intègre les disponibilités des créneaux sur une semaine pour un groupe donné.
5. Écrire une contrainte qui intègre une pause d'une heure et demie minimum chaque jour entre le matin et l'après-midi.
6. Écrire une contrainte qui impose un non recouvrement des créneaux pour une liste de séances en y intégrant une pause minimum de 15 minutes.

5 Contraintes sur plusieurs listes de créneaux

L'objectif de cette section est de mettre en place un ensemble de contraintes permettant de spécifier des intervalles de valeurs à plusieurs listes de créneaux.

1. Écrire une contrainte qui unifie les créneaux de plusieurs groupes correspondant à la même séance.
2. Écrire une contrainte qui intègre la non simultanée des créneaux de l'ensemble des groupes d'un même professeur.

6 Génération d'un emploi du temps

Écrire un programme Prolog qui génère un emploi du temps sur une semaine pour un ensemble de créneaux en utilisant les requêtes et les contraintes précédentes.

Un créneau est représenté par un entier afin de pouvoir utiliser la bibliothèque `clpfd`. Chaque groupe est associé à une liste de créneaux et à une liste de séances. Nous disposons donc de deux listes pour la génération d'un emploi du temps :

1. une liste de listes de créneaux, correspondant aux créneaux de tous les groupes.
2. une liste de listes de séances, correspondant aux séances de tous les groupes.

Pour simplifier le traitement, vous pouvez construire deux listes de même structure, l'une contenant les séances (récupérables à partir du numéro de semaine), l'autre contenant les créneaux (calculés à partir des contraintes).

Exemple : `edt(12,ListeSeances,ListeCreneaux)`

Une réponse possible est :

`ListeSeances=[[seance1,seance2,seance3],[seance1,seance4],...]`

`ListeCreneaux=[[1,6,14],[1,45],...]`

On constate que la séance `seance1`, suivie par deux groupes est synchronisée (premier créneau du lundi matin).

7 Complexité

1. Déterminer la classe de complexité de notre problème.
2. Evaluer une borne sur le nombre de groupes et de séances pour garantir une génération praticable (en temps).

LIVRABLE : Réaliser un tableau comparatif des différentes valeurs du nombre de groupes et de séances en fonction du temps de calcul nécessaire.

8 Améliorations

Selon le même principe que précédemment, proposer une solution afin d'intégrer UNE DES DEUX améliorations suivantes, et implémentez la :

1. Ajouter les données et les traitements nécessaires à la gestion des salles. Vous devrez tenir compte de la taille des salles et de leur type (`cours,td,tp`) afin de proposer en plus des créneaux une affectation d'une salle à chaque séance.
2. Ajouter les données et les traitements nécessaires à la gestion de la disponibilité du matériel de vidéo-projection. Vous devrez modifier les tables existentes et modifier vos contraintes en conséquences.

LIVRABLE : Tous les programmes Prolog dans un seul fichier avec l'extension `*.pl`, accompagnés de la notice d'utilisation.

APPENDICES

A Structure des données

Les données disponibles pour la génération sont les suivantes :

- les promotions, par exemple *cpi1, cpi2, ing1, ing2, ing3*.
- les matières, par exemple *logique, prolog, decidabilite*.
- les professeurs.
- les groupes associés aux informations suivantes :
 - l'effectif du groupe (entier).
 - la promotion à laquelle il appartient.
- les séances associées aux informations suivantes :
 - la matière associée à cette séance.
 - le(s) groupe(s) qui sui(ven)t cette séance.
 - le(s) professeur(s) qui enseigne(nt) cette séance.
 - le type de séance (cours,td,tp).
 - la semaine prévue pour cette séance (de 1 à 52).
 - la durée de cette séance (en minutes).
- les indisponibilités des professeurs associées à :
 - une semaine
 - un jour
 - un intervalle d'indisponibilité (heure début,heure fin)
- les indisponibilités des créneaux associées à :
 - un groupe
 - une semaine
 - un jour
 - un intervalle d'indisponibilité (heure début,heure fin)

B Jeu de test

Pour tester notre application, nous allons planifier la semaine du 19 au 24 octobre 2009 à l'EISTL.

Les matières enseignées sont les suivantes :

TAB. 1: Matières de CPI1

nom
INFORMATIQUE
LV1
ANALYSE
PHYSIQUE
ALGEBRE
FRANCAIS

TAB. 2: Matières de CPI2

nom
INFORMATIQUE
LV1
PHYSIQUE
MATHEMATIQUES
ALGEBRE LINEAIRE LIN
FRANCAIS

TAB. 3: Matières d'ING1

nom
ADA
COMPTABILITE GENERALE
ALGORITHMIQUE1
EXPRES. ORALE-EO
LV2-ESP
LV1
ALGEBRE
ELECTRONIQUE
STATISTIQUES DESCRIPTIVES

TAB. 4: Matière d'ING2

nom
MICROECONOMIE
LOGIQUE COMPUTATIONNELLE
LV2-ESP
LV1
TOPOLOGIE OPTIMISATION
RESEAU TC
Programmation C
INTEGRAT. MESURE
ANALYSE FONCTIONEL.
EDP
STATISTIQUE
PROCESSUS STOCHASTIQUE
CONCEPT. ORIENTE OBJET
GESTION FINANCIERE

La gestion des groupes et des professeurs dépendant du Campus (Cergy ou

Pau), libre à vous de créer les données. Le but est d'être le plus représentatif possible de la réalité.